Accelerating Discrete Diffusion Models with Parallel-In-Time Sampling

Yu Yao
yuyao@g.ecc.u-tokyo.ac.jp
The University of Tokyo

Huanjian Zhou[1]
zhou-huanjian185@g.ecc.u-tokyo.ac.jp
The University of Tokyo

Andi Han
andi.han@sydney.edu.au
The University of Sydney

Wei Huang
wei.huang.vr@riken.jp
RIKEN AIP & The Institute of Statistical Mathematics

Masashi Sugiyama
sugi@k.u-tokyo.ac.jp
RIKEN AIP & The University of Tokyo

Abstract

Discrete diffusion models are widely used for learning and generating discrete distributions. As the generation process is inherently sequential, the acceleration of sampling is of significant importance. In this work, we parallelize the mainstream τ-leaping algorithm for absorbing discrete diffusion in a Continuous-Time Markov Chain (CTMC) framework. By leveraging the continuous-time stochastic integral form of the τ-leaping algorithm and the Picard iteration method, we achieve parallel-in-time sampling acceleration and provide a proof of exponential-factorial convergence for our algorithm. We improve the overall time complexity of τ-leaping under absorbing settings from $\mathcal{O}(d \log S)$ to $\mathcal{O}(\log(d \log S) \cdot \log d)$ with respect to NFE. Empirically, our method shows consistent acceleration across synthetic and real-data settings. The new sampler achieves at most 7–9$\times$ runtime speedup for synthetic distribution, and maintains the same quality with 50% fewer NFE and 1.45–$1.86\times$ runtime speedups in image/text tasks on a single GPU. Our research expands the potential of discrete diffusion models for efficient parallel inference, with broader implications for applications such as molecular structure and language generation.

1. Introduction

Diffusion models over discrete spaces (Austin et al., 2021) have become central to generative modeling for categorical data, with increasing impact across applications including molecular design (Vignac et al., 2022), protein engineering (Gruver et al., 2023), DNA sequence generation under biophysical constraints (Sarkar et al., 2024), and high-fidelity generation of text (Zheng et al., 2023), music (Yang et al., 2023), and images (Lezama et al., 2022). One of the mainstream discrete diffusion is the Absorbing discrete diffusion, which progressively replaces discrete tokens with a special mask or absorbing state, then learns to reconstruct the original data by reversing this corruption process. However, the algorithms underlying the above results are highly sequential and fail to fully exploit contemporary parallel computing resources for modern GPUs. While parallel-in-time algorithms for continuous diffusion models have been extensively studied (Anari et al., 2024; Chen et al., 2024a; Shih et al., 2023; Zhou & Sugiyama, 2025), their counterparts for discrete diffusion models remain largely unexplored. This gap motivates our investigation into the question:

Whether parallelization can fundamentally accelerate sampling for discrete diffusion models?

1: Corresponding author

Accepted to FoGen 2026: Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning, an ICML 2026 workshop (non-archival).

Table 1. Comparisons of our parallel and sequential methods for discrete diffusion sampling given δ-accuracy of the score. Here, S denotes the size of vocabulary.

	Time Complexity	Space Complexity	Metrics
(Ren et al., 2025a, Theorem 4.7)	$\mathcal{O}(\frac{d^2 \log^2(d/\delta^2)}{\delta^2})$	$\mathcal{O}(d)$	KL-divergence
(Conforti et al., 2025, Theorem 3.1.3)	$\mathcal{O}(\frac{dS}{\delta^2})$	$\mathcal{O}(d)$	Total Variation
(Liang et al., 2025a, Theorem 2)	$\mathcal{O}(\frac{dS}{\delta})$	$\mathcal{O}(d)$	KL-divergence
(Dmitriev et al., 2026, Theorem 2)	$\mathcal{O}(\frac{\mathcal{D}(q_0)}{\delta}) \leq \mathcal{O}(\frac{d \log S}{\delta})$	$\mathcal{O}(d)$	KL-divergence
Ours, Theorem 4.10	$\mathcal{O}\left(\log \frac{d \log S}{\delta^2} \cdot \log \frac{d}{\delta}\right)$	$\widetilde{\mathcal{O}}(d^2 S)$	Total Variation

1.1. Our contributions

In this paper, we make significant progress in both theoretical and practical sides. The key contributions of this work are summarized as follows:

- We introduce the *first parallel-in-time* algorithm for τ-leaping methods (Campbell et al., 2022; Gillespie, 2001) for absorbing discrete diffusion inference, namely the Picard τ-leaping method (Algorithm 1).
- We provide well-established theoretical analysis of the method, proving $\mathcal{O}(\log(d \log S) \cdot \log d)$ time complexity compared with $\mathcal{O}(d \log S)$ complexity of mainstream sequential methods with extra $\widetilde{\mathcal{O}}(dS)$ space complexity[2] cost (Theorem 4.10). We summarize the comparison between existing methods and our results in Table 1.
- We verify our method on both synthetic and real-world tasks and achieve substantial improvements in sampling efficiency over various sequential and acceleration methods (Section 5).

1.2. Technical Overview: From Coordinate Parallelism to Time Parallelism

Sampling algorithms for absorbing discrete diffusion can be broadly divided into *Approximate Time-Discretization Methods* and *Exact Simulation Methods*. The typical approximate sampler, τ-leaping sampler, freezes the reverse rates over short time intervals and applies multiple coordinate-level jumps *in parallel* within each step. A general stochastic-integral framework for discrete diffusion was developed by (Ren et al., 2025a), who represent discrete diffusion through Poisson random measures with evolving intensity but its generic τ-leaping bounds are conservative for absorbing processes.

Many recent work focuses specifically on the absorbing dynamics. (Huang et al., 2025) study the complexity of masked discrete diffusion and propose Mask-Aware Truncated Uniformization (MATU), which leverages the fact that masked tokens cannot be unmasked multiple times and adapts the truncation of outgoing rates to the number of remaining masked coordinates. (Liang et al., 2025a) gives the first rigorous convergence guarantees for absorbing τ-leaping under bounded score estimates. (Conforti et al., 2025) provide sharper non-asymptotic bounds for masked and random-walk discrete diffusion, using the evolution and monotonicity of discrete scores to avoid strong boundedness assumptions. (Dmitriev et al., 2026) analyze discrete diffusion through information-theoretic quantities, proving sharp upper bound that intrinsically depends on the structural properties of the target distribution. Together, these results highlight that masking diffusion has exploitable structure beyond generic finite-state CTMCs, and show that sequential masking samplers can achieve at most linear complexity up to logarithmic factors.

Compared with previous sequential methods and analysis, our method keeps the same fine grid but approximate the sequential chain through parallelism along the time horizon. We group many fine intervals into a larger block and solve the whole block transition by Picard iteration. In each Picard round, all microsteps in the block are evaluated in parallel from the previous round's trajectory using the same block randomness generated at the beginning of the algorithm. Compared to the Picard methods for continuous diffusion models (Chen et al., 2024a; Shih et al., 2023; Zhou & Sugiyama, 2024), we additionally apply a first-hitting truncation inside the block for masking diffusion, preserving the absorbing structure while making the block update compatible with parallel prefix operations. By proving an

[2]We note, in this paper, that the space complexity refers to the number of words (Cohen-Addad et al., 2023) instead of the number of bits (Goldreich, 2008) to denote the approximate required storage.

exponential-factorial contraction of the Picard error, we derive $O(\log d)$ level estimation for both number of blocks and Picard iterations, which eventually leads to $\mathrm{poly}\log(d)$ time complexity.

1.3. Other Related Work

Exact simulation of reverse process with learned score. Exact simulation methods, such as uniformization for uniform-rate chains (Chen & Ying, 2024) and the First-Hitting Sampler (FHS) (Liang et al., 2026; Zheng et al., 2024) for absorbing chains, simulate the reverse CTMC by separating the sampling of jump times from the sampling of jump destinations. This removes time-discretization error and, in the absorbing case, can directly exploit the fact that each coordinate is unmasked at most once. In the ideal exact-score setting, such methods provide unbiased simulation of the learned reverse with d steps sampling. However, exact simulation does not necessarily lead to better generation quality in practice. Recent empirical results even show that exact simulation can underperform approximate solvers despite having no discretization error (Ren et al., 2025b). One possible reason is that exact samplers concentrate many jumps near the terminal phase of the reverse process, precisely where score estimation is most singular and inaccurate. This motivates continued study of approximate samplers, especially the τ-leaping method, whose discretization can act as a stabilizing numerical approximation of learned-score errors.

Other acceleration strategies A large body of empirical work accelerates discrete diffusion by changing the decoding strategy rather than the time integration scheme. Parallel decoding methods unmask multiple tokens per iteration using confidence, entropy, margin, or related criteria, thereby exploiting spatial or token-level parallelism (Chang et al., 2022; Ringel et al., 2026; Xie et al., 2025; Wu et al., 2025; Yu et al., 2025). These methods are effective in practice, but the sampling horizon still consists of a sequence of denoising rounds. Other approaches modify the generation order or introduce auxiliary planning modules. DDPD (Liu et al., 2024) separates generation into a planner and a denoiser, allowing the model to decide which corrupted positions should be refined next; related path-planning methods study how the unmasking order affects sample quality (Peng et al., 2025). High-order solvers (Ren et al., 2025b) extend ideas such as trapezoidal integration to jump processes, reducing discretization error and permitting larger steps under the same computation budget.

2. Preliminaries on Discrete Diffusion Models

In this section, we briefly introduce the background knowledge about the mathematical framework of discrete diffusion and its properties under mask settings. Throughout the sampling and theoretical analysis, t denotes the forward noising time where $t = 0$ is the clean-data endpoint, and the early-stopped reverse sampler runs from T down to $\eta > 0$.

2.1. Continuous-Time Markov Chains and Poisson Integrals

In discrete diffusion models, the forward process is a continuous-time Markov chain (CTMC) $(\boldsymbol{x}_t)_{t\in[0,T]}$ on a finite state space $\mathcal{X}$. Let $\boldsymbol{p}_t \in \Delta^{|\mathcal{X}|}$ denote the law of $\boldsymbol{x}_t$ as a column vector. The forward equation is $\frac{\mathrm{d}\boldsymbol{p}_t}{\mathrm{d}t} = \boldsymbol{Q}_t\,\boldsymbol{p}_t$, $\boldsymbol{Q}_t = \big(\boldsymbol{Q}_t(y,x)\big)_{x,y\in\mathcal{X}}$, where $\boldsymbol{Q}_t$ is a rate matrix with (i) $\boldsymbol{Q}_t(x,y) \geq 0$ for $x \neq y$ and (ii) $\boldsymbol{Q}_t(x,x) = -\sum_{y\neq x} \boldsymbol{Q}_t(y,x)$. We write $\widetilde{\boldsymbol{Q}}_t := \boldsymbol{Q}_t - \mathrm{diag}(\boldsymbol{Q}_t)$ for the off-diagonal part. The time-reversed (backward) process $\big(\overleftarrow{\boldsymbol{x}}_t\big)_{t\in[0,T]}$, with $\overleftarrow{*}_t := *_{T-t}$, is again a CTMC with law $\overleftarrow{\boldsymbol{p}}_s$ and generator $\overleftarrow{\boldsymbol{Q}}_t$ satisfying (Kelly, 2011) $\frac{\mathrm{d}\,\overleftarrow{\boldsymbol{p}}_t}{\mathrm{d}t} = \overleftarrow{\boldsymbol{Q}}_t\,\overleftarrow{\boldsymbol{p}}_t$, where for $x \neq y$, $\overleftarrow{\boldsymbol{Q}}_t(y,x) = \frac{\overleftarrow{\boldsymbol{p}}_t(y)}{\overleftarrow{\boldsymbol{p}}_t(x)}\,\boldsymbol{Q}_t(x,y)$, and $\overleftarrow{\boldsymbol{Q}}_t(x,x) = -\sum_{y'\neq x} \overleftarrow{\boldsymbol{Q}}_t(y',x)$, and $\overleftarrow{\boldsymbol{Q}}_t := \boldsymbol{Q}_{T-t}$.

According to Proposition 3.2 in (Ren et al., 2025a), discrete diffusion models can also be interpreted as stochastic integrals with Poisson random measure. The forward process in discrete diffusion models can thus be represented by the following stochastic integral: $\boldsymbol{x}_t = \boldsymbol{x}_0 + \int_0^t \int_{\mathbb{D}} \nu N[\lambda](\mathrm{d}t, \mathrm{d}\nu)$, where the intensity λ is defined as $\lambda_t(\nu,\omega) = \overleftarrow{\boldsymbol{Q}}_t(x_{t^-}(\omega)+\nu, x_{t^-}(\omega))$ if $x_{t^-}(\omega)+\nu \in \mathcal{X}$ and 0 otherwise. Here, the outcome $\omega \in \Omega$ and x_{t^-} denotes the left limit of the càdlàg process x_t at time t with $x_{0^-} = x_0$. We will also omit the variable ω, should it be clear from context.

The backward process in discrete diffusion models can also be represented similarly as

$$\boldsymbol{y}_t = \boldsymbol{y}_0 + \int_0^t \int_{\mathbb{D}} \nu N[\mu](\mathrm{d}s, \mathrm{d}\nu), \tag{1}$$

where the intensity μ is defined as $\mu_t(\nu,\omega) = \overleftarrow{s}_t(\boldsymbol{y}_{t^-}, \boldsymbol{y}_{t^-}+\nu)\overleftarrow{\boldsymbol{Q}}_s(\boldsymbol{y}_{s^-}, \boldsymbol{y}_{s^-}+\nu)$. During inference, $\widehat{\mu}$ defined by

replacing the true score $\boldsymbol{s}_t$ with the neural network estimated score $\widehat{\boldsymbol{s}}_t$ is used.

2.2. Masking Diffusion

In this work, we focus on absorbing, or masking discrete diffusion on the clean state space $\mathcal{X}_0 = [S]^d$ and the masked state space $\mathcal{X}_M = ([S] \cup \{\mathrm{MASK}\})^d$. The forward process independently replaces each clean coordinate by MASK and keeps MASK absorbing. Let $\beta_t \geq 0$ be the forward masking rate (i.e. noise schedule) and $\alpha_t := \exp\Big(-\int_0^t \beta_s \, ds\Big)$. Then, for a clean sample $x_0 \in [S]^d$, the forward marginal is $q_t(z \mid x_0) = \prod_{i=1}^d \Big[\alpha_t \, \mathbf{1}\{z_i = x_{0,i}\} + (1-\alpha_t)\, \mathbf{1}\{z_i = \mathrm{MASK}\}\Big], z \in \mathcal{X}_M$. Equivalently, the forward CTMC generator only allows transitions from ordinary tokens to MASK: $Q_t(z, z^{i\to\mathrm{MASK}}) = \beta_t \, \mathbf{1}\{z_i \neq \mathrm{MASK}\}$, $Q_t(z,z) = -\beta_t \, |\{i : z_i \neq \mathrm{MASK}\}|$. Here $z^{i\to\mathrm{MASK}}$ denotes the state obtained by replacing coordinate i of z with MASK.

The reverse process has a particularly simple structure. Let $O(z) := \{i : z_i \neq \mathrm{MASK}\}$ be the observed coordinates of a partially masked state z. For a masked coordinate $i \notin O(z)$, define the clean-data posterior $\pi_i(c \mid z) := \mathbb{P}_{X_0 \sim p_{\mathrm{data}}}\big(X_{0,i} = c \,\big|\, X_{0,O(z)} = z_{O(z)}\big), c \in [S]$. Because the masking channel treats all hidden token values identically, this posterior is independent of the corruption time t. The reverse rate from z to the state $z^{i\leftarrow c}$, obtained by replacing MASK at coordinate i with token c, is

$$\mu_t(z^{i\leftarrow c}, z) = \beta_t \frac{q_t(z^{i\leftarrow c})}{q_t(z)} = \frac{\beta_t \alpha_t}{1-\alpha_t}\, \pi_i(c \mid z) = a_t \, \pi_i(c \mid z), \qquad a_t := \frac{\beta_t \alpha_t}{1-\alpha_t}.$$

Thus the reverse rate separates into a scalar time factor a_t and a time-independent conditional token posterior. A learned score model is used to approximate this posterior, giving rates of the form $\widehat{\mu}_t(z^{i\leftarrow c}, z) = a_t \, \widehat{p}_i^{\,\theta}(c \mid z)$ and $\sum_{c\in[S]} \widehat{p}_i^{\,\theta}(c \mid z) = 1$. Finally, masking diffusion has a first-hitting structure. Along reverse sampling, a coordinate remains masked until its first reveal time $\tau_i \in [\eta, T]$, and then stays fixed as $X_i(t) = \mathrm{MASK}$ for $t > \tau_i$ and $X_i(t) = c_i$ for $t \leq \tau_i$. This absorbing first-hitting property distinguishes masking diffusion from uniform or random-walk discrete diffusion, where coordinates may jump repeatedly, and is the structural property exploited by our Picard sampler.

3. Parallel Sampling for Absorbing Discrete Diffusion Models: Algorithm 1

In this section, we introduce the parallel-in-time τ-leaping algorithm for absorbing discrete diffusion models. The key idea is to extend the Picard iteration from continuous diffusion models to the stochastic integral form of τ-leaping in discrete state spaces.

Discretization Scheme. Following the time convention above, the reverse sampler is executed on the forward-noise interval $[\eta, T]$ in decreasing time order. We use a large noise-time grid $T = t_0 > t_1 > \cdots > t_N = \eta$. For block n, the interval $[t_{n+1}, t_n]$ is divided into M fine cells using $\tau_{n,m} = t_n - m\epsilon_n, \quad \epsilon_n = \frac{t_n - t_{n+1}}{M}, \quad m = 0, \ldots, M$. The positive number ϵ_n is the fine-step width used in the Poisson proposal mean.

Picard update. When describing the standard τ-leaping algorithm with the form in equation 1, the main update step in Picard iteration can be described as

$$\widehat{y}^{(k+1)}_{\tau_{n,m}} = \widehat{y}_{t_n} + \sum_{j=0}^{m-1} \Bigg(\sum_{y' \in \mathbb{X}} \big(y' - \widehat{y}^{(k)}_{\tau_{n,j}}\big) \cdot \mathcal{P}\big(\widehat{\mu}^{\theta}_{\tau_{n,j}}\big(y' \mid \widehat{y}^{(k)}_{\tau_{n,j}}; \xi_n\big) \cdot \epsilon_n\big) \Bigg). \tag{2}$$

where $\widehat{y}^{(k)}_{\tau_{n,j}}$ is the sample state a $t = t_n - j\epsilon$ in the k-th Picard iteration. $\mathcal{P}(\widehat{\mu}^{\theta}_{\tau_{n,j}}(y' | \widehat{y}^{(k)}_{\tau_{n,j}}); \xi_n)$ denotes the number of jumps from $\widehat{y}^{(k)}_{\tau_{n,j}}$ to y' during the small time interval ϵ. In the block-wise parallel algorithm, the computation for each block starts from its initial state $\widehat{y}_{t_n}$ which is also the terminal value of the last block. Check Lines 4–7 in Algorithm 1 for details. The sampler admits a coordinate-token factorization (Campbell et al., 2022): each fine cell proposes token updates over d coordinates and S vocabulary entries, giving $O(dS)$ local proposal cost rather than S^d dependence. The Picard update then replaces the serial dependence on the previous fine step by dependence on the full trajectory from the previous iteration, and all proposals inside a block can be computed in parallel. After applying first-hitting truncation, the next block trajectory is reconstructed from the block start state using a parallel prefix sum over the

Algorithm 1 Parallel τ-Leaping Algorithm for Discrete Diffusion Model Sampling

Input: $\widehat{y}_{t_0} \sim q_T$, large time grid $(t_n)_{n\in[0,N]}$ with $t_0 = T$, $t_N = \eta$, and $t_{n+1} < t_n$; small time grid $(\tau_{n,m})_{n\in[0,N-1],m\in[0,M]}$ with $\tau_{n,0} = t_n$, $\tau_{n,M} = t_{n+1}$ and positive step size $\epsilon = t_n - t_{n+1}$ divided by M, i.e. $\tau_{n,m} = t_n - m\epsilon$; Picard depth K_p; intensity $\widehat{\mu}^\theta_s$; score estimate $\widehat{s}^\theta_t$; pre-sampled random seeds $(\xi_n)_{n\in[0,N-1]}$; First-hitting truncation operation $\mathrm{FHT}(\cdot)$.
Output: An early-stopped sample $\widehat{y}_{t_N}$ with $t_N = \eta$.

for $n = 0$ to $N-1$ **do**
  Initial Guess: $\widehat{y}^{(0)}_{\tau_{n,m}} \leftarrow \widehat{y}_{t_n}$ for all $m \in [0, M]$
  for $k = 0$ to $K_p - 1$ **do**
    for $j = 0$ to $M-1$ **in parallel do**
      $\widehat{\mu}^\theta_j \leftarrow \widehat{\mu}^\theta_{\tau_{n,j}}\big(\cdot\,|\widehat{y}^{(k)}_{\tau_{n,j}}\big)$
      $J_{j,y'} \sim \mathcal{P}(\widehat{\mu}^\theta_j(y')\cdot\epsilon;\xi_n)$ for all $y' \in \mathcal{X}_{neighbor}$
      $\Delta\widehat{y}^{(k)}_j = \sum_{y'\in\mathcal{X}_{neighbor}} \big(y' - \widehat{y}^{(k)}_{\tau_{n,j}}\big)\cdot J_{j,y'}$
    end for
    $\{\Delta\widehat{y}^{(k)}_j\}_{j\in[0,M-1]} \leftarrow \mathrm{FHT}\Big(\{\Delta\widehat{y}^{(k)}_j\}_{j\in[0,M-1]}\Big)$
    for $m = 1$ to M **in parallel do**
      $\widehat{y}^{(k+1)}_{\tau_{n,m}} \leftarrow \widehat{y}_{t_n} + \sum_{j=0}^{m-1}\Delta\widehat{y}^{(k)}_j$
    end for
  end for
  $\widehat{y}_{t_{n+1}} \leftarrow \widehat{y}^{(K_p)}_{\tau_{n,M}}$
end for

truncated jump vectors. With pre-sampled shared random seeds ξ_n, the Poisson sampling performs as a deterministic map for convergence in each block. Since multi-time jump at the same time on one coordinate is not well-defined, such case is also truncated (Liang et al., 2025b).

First-hitting truncation. First-hitting truncation is designed for absorbing settings in Algorithm 2 (check Appendix D for details). Given the local proposal events in a block prefix before the trajectory reconstruction, the first-hitting operation returns the state obtained by applying, for each coordinate, only its earliest proposed token.

4. Theoretical Guarantees

In this section, we provide a proof of the Picard Convergence and an approximated error bound of our algorithm under specific definition and assumptions.

4.1. Assumptions

We now introduce the definitions and assumptions used for the proof of theoretical guarantees. Definition 4.1 and 4.3 provide necessary measure for evaluating the random proposal and error propagation. Assumption 4.2 is a Dobrushin-style perturbation controlling condition, which can be considered as an analogue to the Lipschitz condition in continuous diffusion. Assumption 4.4 and 4.5 are inherited from the absorbing τ-leaping convergence theory in (Liang et al., 2025a) which allow us to isolate the additional error introduced by parallelizing the sequential sampler.

Definition 4.1 (Fine cells and local proposal events). Let $\mathcal{Q}_d$ be the set of fine-grid cells covering the early-stopped interval $[\eta, T]$. For each $q \in \mathcal{Q}_d$, let Δ_q be its fine-step width, t_q its representative time, and $a_q = a(t_q)$ be the total unmasking rate of one active masked coordinate. Define $\lambda_q = \Delta_q a_q$, $\rho_q = \lambda_q e^{-\lambda_q} \le \lambda_q = \Delta_q a_q$. Here ρ_q is the one-proposal probability at one active coordinate in cell q.

For an input state z and cell randomness ω_q, let $C_q(z,\omega_q) \subseteq [d]\times[S]$ be the set of valid local proposal events generated in cell q. An event $(i,c) \in C_q(z,\omega_q)$ means that coordinate i is locally proposed as token c.

Assumption 4.2 (Normalized event switching). There exists a constant $L_\star > 0$ such that, for every fine cell q and all relevant states $z, z' \in \mathcal{X}_M$,

$$\mathbb{E}_{\omega_q}\left[|C_q(z,\omega_q)\triangle C_q(z',\omega_q)|\right] \le L_\star\rho_q d_H(z,z').$$

This assumption works as a model-sensitivity condition in the Picard proof. The factor ρ_q captures how likely cell q is to produce a local proposal at all, and $L_\star$ controls how strongly a token-level perturbation of the input context can change the local proposal event set, whose constant property is also verified in Appendix E.1 and Figure 2.

Connection to score sensitivity. The event-switching condition can be viewed as a proposal-level analogue of a score Lipschitz condition. For masking diffusion, the reverse rate factorizes as $\widehat{\mu}_{q,i,c}(z) = a_q \widehat{p}_i^\theta(c \mid z)$. If the normalized token posterior satisfies $\sum_i \left\| \mathbf{1}\{z_i = \text{mask}\}\widehat{p}_i^\theta(\cdot \mid z) - \mathbf{1}\{z_i' = \text{mask}\}\widehat{p}_i^\theta(\cdot \mid z')\right\|_1 \le L_{\text{score}} d_H(z, z')$.

Then, under the per-cell Poisson proposal coupling, the expected candidate-event switching obeys $\mathbb{E}_{\omega_q}\left[|C_q(z,\omega_q) \triangle C_q(z',\omega_q)|\right] \le C\rho_q L_{\text{score}} d_H(z,z')$, where C is a constant depending on the way of coupling. Thus our normalized event-switching assumption is a direct algorithm-level consequence of a Dobrushin-type score sensitivity bound, with the proposal probability ρ_q separating temporal sparsity from model sensitivity.

Definition 4.3 (Propagation masses). For each fine cell, set $b_q = L_\star \rho_q$. If block n contains the fine cells $\mathcal{B}_n$, define the block and global propagation mass by

$$B_n = \sum_{q \in \mathcal{B}_n} b_q, \quad G_d = \sum_{q \in \mathcal{Q}_d} b_q, \quad G_d = \sum_{n=1}^{N} B_n.$$

The quantity B_n controls the Picard difficulty of block n, and G_d controls the total propagation mass over the whole sampling interval.

Assumption 4.4 (Score Estimation Error). Let $u_0 = T > u_1 > \cdots > u_{N_{\text{fine}}} = \eta$ be the serial fine grid. The estimated score satisfies $\sum_{\ell=0}^{N_{\text{fine}}-1} (u_\ell - u_{\ell+1}) \mathcal{L}_{\text{SE}}(\widehat{s}_{u_\ell}) \le \varepsilon_{\text{score}}$.

Assumption 4.5 (Bounded Score Estimate). There exists $M_{\text{score}} > 0$ such that for all $x, y \in [S]^d$ with $Q_{u_\ell}(y, x) > 0$, $|\log \widehat{s}_{u_\ell}(y, x)| \le \log M_{\text{score}}$.

4.2. Picard Endpoint Convergence and Block-wise Complexity

We first present the theorem for Picard convergence.

Theorem 4.6 (Picard endpoint convergence). *Let $Y^{(k)} = Z_M^{(k)}$ be the state at the endpoint of a time block. Define the adjacent Picard residual under Hamming distance as $e_k(m) = \mathbb{E}\left[d_H\left(Z_m^{(k+1)}, Z_m^{(k)}\right)\right]$. Let $B = \sum_{r=0}^{M-1} b_r$ and $E_0 = \max_{0 \le m \le M} e_0(m)$. Under Assumption 4.2, the block endpoint iterates converge to a limit $Y^{(\infty)}$ with*

$$\mathbb{E}\left[d_H\left(Y^{(K)}, Y^{(\infty)}\right)\right] \le E_0 e^B \frac{B^K}{K!}$$

for every $K \ge 0$. For block n, the same statement holds with $B = B_n$. One may always take $E_0 \le d$.

The theorem shows that the iteration residual of discrete diffusion under Hamming distance performs exponential-factorial contraction compared with typical exponential contraction in continuous diffusion (Chen et al., 2024a). Please refer to Appendix B.1 for the proof.

Remark 4.7. The exponential-factorial contraction in Theorem 4.6 is not exclusive to the discrete diffusion. It is actually the discrete analogue of the classical Volterra expansion for Picard iterations in continuous case. Please refer to Appendix B.3 for details.

Corollary 4.8 (Uniform-block Picard NFE). *Let $H := T - \eta$ be the early-stopped sampling horizon. Choose a constant physical block width $h_0 = O(1) > 0$ independent of d and ε_{pic}, and partition $[\eta, T]$ into $N = \left\lceil \frac{H}{h_0} \right\rceil$ uniform physical-time blocks. Under Assumption 4.2, given $T = O(\log(d \log S/\varepsilon_{\text{pic}}))$, let $B_{\max} = \max_n B_n$. Then the choice*

$$K_p = \left\lceil \max\left\{ 2eB_{\max}, \frac{1}{\log 2} \log \frac{Nde^{B_{\max}}}{\varepsilon_{\text{pic}}} \right\} \right\rceil$$

ensures that the Picard endpoint TV error at the final block $\text{TV}(\mathcal{L}(Y_N^{(K_p)}), \mathcal{L}(Y_N^{(\infty)}))$ is at most ε_{pic}, where $\mathcal{L}(\cdot)$ is the law of variables. Consequently,

$$K_p = O\left(\log(d\varepsilon_{\text{pic}}^{-1})\right), \quad \text{NFE}_{\text{Picard}} = NK_p = O\left(\log(d\varepsilon_{\text{pic}}^{-1}) \cdot \log(d\varepsilon_{\text{pic}}^{-1} \log S)\right).$$

Here, we prove that an ε-level block endpoint TV error can be achieved with $O(\log^2 d)$ total NFE under *constant level physical block width*, which implies the true block-wise parallel sampling. Such result is also independent to the fine-grid partition and discretization error. Please refer to Appendix B.2 for the proof.

4.3. Total Error Decomposition and Global Complexity

We now connect the Picard endpoint theorem to a sequential convergence theorem for absorbing discrete diffusion. We take the absorbing τ-leaping sampler in (Liang et al., 2025a) as the serial reference that our Picard block iteration parallelizes. Let μ_0 be the clean data distribution, $\mu_\eta^\star$ the exact absorbing reverse law at early stopping time η, and A_η the deterministic completion map used after early stopping. Let $\nu_{\text{seq}}^{\theta,\Delta}$ be the output law of the absorbing serial τ-leaping sampler on the fine grid, and let $\nu_{\text{pic}}^{\theta,\Delta,K_p}$ be the output law of the Picard sampler with depth K_p. We define $E_{\text{term}} := \mathrm{TV}(\mu_0, A_\eta \# \mu_\eta^\star)$.

Absorbing serial reference (Liang et al., 2025a) The absorbing τ-leaping theorem gives a KL bound of the form $\mathrm{KL}(\mu_\eta^\star \| \nu_{\text{seq}}^{\theta,\Delta}) \le E_{\text{absTL}}$, where, up to universal and logarithmic constants,

$$E_{\text{absTL}} := de^{-T}\log S + \varepsilon_{\text{score}} + \frac{dS\big(T + \log(M_{\text{score}}\eta^{-1})\big)\big(T + \log \eta^{-1}\big)^2}{N_{\text{fine}}}.$$

Here N_{fine} is the number of serial fine steps, $\varepsilon_{\text{score}}$ is the score entropy error, and M_{score} is the bounded-score constant. The same result gives the early-stopping control $E_{\text{term}} \lesssim d\eta$.

Proposition 4.9 (Total TV error with absorbing τ-leaping reference). *Assume the absorbing serial τ-leaping bound above and the Picard endpoint bound in Theorem 4.6. Then*

$$\mathrm{TV}(\mu_0, A_\eta \# \nu_{\text{pic}}^{\theta,\Delta,K_p}) \le E_{\text{term}} + \sqrt{\frac{1}{2}E_{\text{absTL}}} + \sum_{n=1}^{N} E_{n,0} e^{B_n} \frac{B_n^{K_p}}{K_p!}.$$

The proposition works as a bridge connecting our Picard iteration error with the standard sequential τ-leaping reference and its original truncation/discretization error. The key idea is a direct application of the triangle and Pinsker's inequality for total variation. Please refer to Appendix C.1 for the proof.

Theorem 4.10 (Global complexity). *Let the target total variation error be ε_{tot}. Under Assumption 4.2, 4.4, 4.5, suppose the score error is controlled at order $\varepsilon_{\text{tot}}^2$, the global time and space complexity of Picard method with constant physical block width are*

$$N_{\text{block}} K_p = O\Big(\log \frac{d\log S}{\varepsilon_{\text{tot}}^2} \cdot \log \frac{d}{\varepsilon_{\text{tot}}}\Big), \quad dM = O\Big(d\frac{N_{\text{fine}}}{N_{\text{block}}}\Big) = \widetilde{O}\left(\frac{d^2 S}{\varepsilon_{\text{tot}}^2}\right)$$

where $\eta = \Theta(\varepsilon_{\text{tot}}/d)$, $T = O\left(\log(d\varepsilon_{\text{tot}}^{-2}\log S)\right)$, $N_{\text{fine}} = \widetilde{O}\left(dS\varepsilon_{\text{tot}}^{-2}\right)$, $N_{\text{block}} = O\left(\log(d\varepsilon_{\text{tot}}^{-2}\log S)\right)$, $K_p = O\left(\log(d\varepsilon_{\text{tot}}^{-1})\right)$.

Here, we propose the full parameter schedule for our Picard method under ε_{tot}-level TV error, achieving $O(\log^2 d)$ level time complexity and $\widetilde{O}\left(d^2 S\right)$ level space complexity. Please refer to Appendix C.2 for proof details.

Remark 4.11. The preceding theorem uses constant physical block width, which balances the critical path and memory. If hardware limitations are ignored, one may even place the entire time horizon into a single block as $N_{\text{block}} = 1$, which leads to even $O(\log d)$ level complexity.

5. Experiments

In this section, we empirically evaluate the performance of our parallel τ-leaping algorithm against sequential τ-leaping and related acceleration methods.

5.1. 2D Toy Experiments

We first conduct 2D oracle experiments on sampling 8×8 chessboard distribution and 32×32 Circle distributions with $N = 40, M = 80$ under different Picard depth to verify the convergence of Picard iteration. The results in Table 2

Table 2. Performance comparison on synthetic data.

Alg.	K_p	Chessboard			Circle		
		Runtime (s)	KL Divergence↓	NFE	Runtime (s)	KL Divergence↓	NFE
Seq.	/	3.79 ± 0.08	0.0068 ± 0.0010	3200	4.77 ± 0.21	0.1160 ± 0.0094	3200
Par.	2	0.12 ± 0.06	0.0491 ± 0.0029	80	0.31 ± 0.05	0.1335 ± 0.0060	80
	4	0.22 ± 0.07	0.0233 ± 0.0024	160	0.58 ± 0.04	0.1146 ± 0.0072	160
	6	0.31 ± 0.04	0.0093 ± 0.0017	240	0.85 ± 0.03	0.1118 ± 0.0073	240
	8	0.40 ± 0.06	0.0054 ± 0.0012	320	1.12 ± 0.04	0.1121 ± 0.0068	320
	10	0.53 ± 0.07	0.0056 ± 0.0014	400	1.37 ± 0.05	0.1119 ± 0.0059	400

Table 3. Oracle synthetic quality-matched scaling.

d	Seq NFE	Blocks	M	Para NFE	NFE speedup	Wall speedup	Seq KL	Para KL
256	288	48	6	96	3.0×	1.28×	0.0389	0.0418
512	540	54	10	108	5.0×	2.06×	0.0212	0.0227
1024	1080	60	18	120	9.0×	4.03×	0.0141	0.0164
2048	2048	64	32	128	16.0×	8.73×	0.0081	0.0097
4096	4104	72	57	144	28.5×	12.45×	0.0045	0.0049

show rapid convergence of our algorithm with significant NFE and runtime acceleration. More detailed settings are demonstrated in Appendix E.2.

5.2. Dimensional Scaling

We next evaluate whether the Picard sampler exhibits favorable scaling with the sequence dimension in a controlled oracle setting. We test whether a growing number of sequential Picard blocks is sufficient to match a serial fine-grid reference as d increases. We use a block-product two-mode distribution on binary sequences. Each sample of length d is partitioned into groups of size $g = 8$, and each group independently follows $q_g(y) = \frac{1-\alpha}{2}\delta_{0^g}(y) + \frac{1-\alpha}{2}\delta_{1^g}(y) + \alpha 2^{-g}$ with $\alpha = 0.05$. Table 3 and Figure 1 show the quality-matched scaling behavior. As d increases, the serial reference requires a linearly growing fine grid, while the selected Picard schedule uses a much more slowly growing number of blocks, closely following the logarithmic reference. With fixed $K_p = 2$, this leads to a rapidly increasing critical-path NFE speedup, and the wall-clock measurements follow the same qualitative trend while maintaining near-serial KL quality. More visualization is in Appendix E.2.

5.3. Image Generation

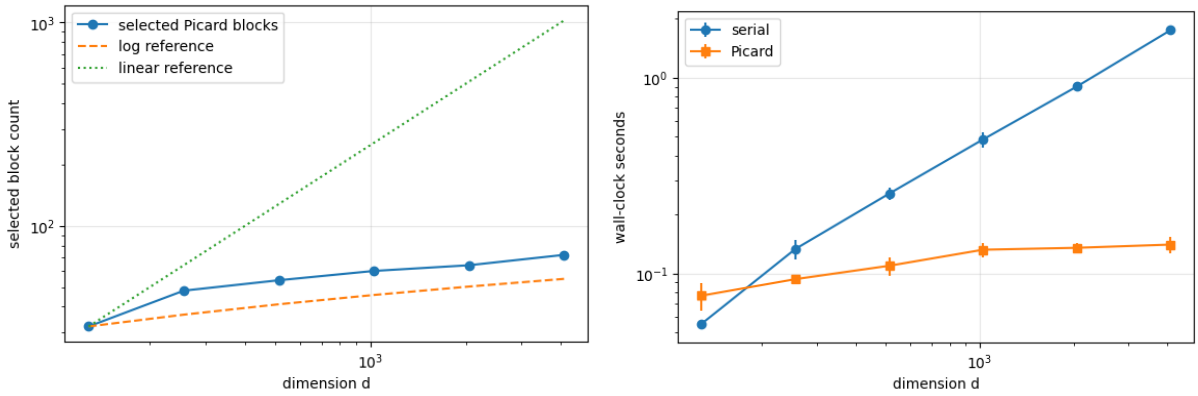
The experiment utilized a MaskGiT-based score model (Besnier & Chen, 2023; Chang et al., 2022) pretrained on ImageNet (Deng et al., 2009). We compared the performance of both sequential and parallel τ-leaping when generating 256×256 resolution images and evaluated the Fréchet Inception Distance (FID) score based on 20k samples. We use classifier-free guidance with guidance scale $w = 3$ to match the settings of baseline methods and verify the ability of combining our algorithm with inference-time control methods. The result in Table 4 shows that our parallel method achieves equivalent quality with half of the NFE. Check Appendix E.3 for more generated sample images.

Table 4. Comparison in image generation.

Method	**NFE**	**FID↓**
FHS	64	14.31
Parallel Decoding	64	13.07
τ-leaping	64	7.56
θ-Trapezoidal	64	6.59
Parallel (Ours)	32	6.82

5.4. Text Generation

The experiment utilized an RADD-based score model (Ou et al., 2024) pretrained on the OpenWebText dataset (Gokaslan & Cohen, 2019) which has GPT-2-level text generation capabilities (Radford et al., 2019). We compared the performance of both sequential and parallel τ-leaping when generating 1024-token texts with vocabulary $S = 50258$, and evaluated the average generative perplexity score with 1024 samples. The results in Table 5 show that our parallel method achieves better perplexity with fixed NFE, and actually achieves the same perplexity level with only half of the



(a) Selected Picard block count (b) Serial vs. Picard wall-clock

Figure 1. Visualization for the scaling experiment.

Table 5. Generative perplexity of texts generated by different sampling algorithms on GPT-2 large.

Method	NFE = 32	NFE = 64	NFE = 128	NFE = 256	NFE = 512	NFE = 1024
FHS	188.653	140.420	124.335	111.959	113.854	110.946
Tweedie τ-leaping	160.466	108.431	83.922	69.745	53.922	43.451
τ-leaping	94.918	67.544	52.384	41.880	35.498	30.762
θ-Trapezoidal	87.895	68.587	50.226	39.119	32.886	25.309
Parallel (Ours)	**82.439**	**50.858**	**40.701**	**32.510**	**25.876**	**22.099**

NFE compared with other sequential and high-order solvers. Please refer to Appendix E.3 for more comparison results with DDPD (Liu et al., 2024).

5.5. Runtime Acceleration

In addition to quality metrics such as FID and PPL, we measure the actual sampling wall-clock time of our Picard sampler against the time-sequential τ-leaping baseline. For a fair comparison, we match the quality-oriented sampling configuration and report pure sampling time, excluding the final image decoding overhead in the image experiment. The speedup is computed as the ratio between the serial runtime and the Picard runtime. The Picard parameters in both two tasks are set as $M = 4,\ K_p = 2$. Table 6 indicates that the NFE reduction translates into practical single-GPU acceleration, although the wall-clock gain is smaller than the ideal NFE ratio due to relatively heavy memory traffic. Based on the research in continuous diffusion (Shih et al., 2023), we expect better results in multi-GPU sampling.

Table 6. Runtime comparison between sequential τ-leaping and Picard sampler.

Task	Device	Serial NFE	Picard Block	Picard NFE	Time (serial / Picard)	Speedup
Text	H100	128	$N = 32$	64	1.652s / 1.139s	1.45×
Image	RTX 4090	64	$N = 16$	32	1.603s / 0.860s	1.86×

6. Discussion and Conclusion

In this work, we proposed a parallel-in-time τ-leaping sampler for absorbing discrete diffusion based on Picard iteration, achieving $O(\log d)$ time complexity with TV-based error analysis and substantial improvement of NFE and runtime speed in experiments. Future work includes solving our method limitation by developing a sharper KL-level analysis of the Picard parallelization error and designing more memory-efficient variants, such as sliding-window Picard updates or adaptive block partitioning with multi-GPU experiments. Combining parallel-in-time sampling with improved serial solvers or sparse vocabulary computation may further improve scalability on large-scale discrete generation.

References

Anari, N., Chewi, S., and Vuong, T.-D. Fast parallel sampling under isoperimetry. *Proceedings of Thirty Seventh Conference on Learning Theory*, pp. 161–185, 2024.

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and Van Den Berg, R. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Besnier, V. and Chen, M. A pytorch reproduction of masked generative image transformer. *arXiv preprint arXiv:2310.14400*, 2023.

Campbell, A., Benton, J., De Bortoli, V., Rainforth, T., Deligiannidis, G., and Doucet, A. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

Chang, H., Zhang, H., Jiang, L., Liu, C., and Freeman, W. T. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11315–11325, 2022.

Chao, C.-H., Sun, W.-F., Liang, H., Lee, C.-Y., and Krishnan, R. G. Beyond masked and unmasked: Discrete diffusion models via partial masking. *arXiv preprint arXiv:2505.18495*, 2025.

Chen, H. and Ying, L. Convergence analysis of discrete diffusion model: Exact implementation through uniformization. *arXiv preprint arXiv:2402.08095*, 2024.

Chen, H., Ren, Y., Ying, L., and Rotskoff, G. Accelerating diffusion models with parallel sampling: Inference at sub-linear time complexity. *Advances in Neural Information Processing Systems*, 37:133661–133709, 2024a.

Chen, Z., Yuan, H., Li, Y., Kou, Y., Zhang, J., and Gu, Q. Fast sampling via discrete non-markov diffusion models with predetermined transition time. *Advances in Neural Information Processing Systems*, 37:106870–106905, 2024b.

Cohen-Addad, V., Woodruff, D. P., and Zhou, S. Streaming Euclidean k-median and k-means with $o(\log n)$ Space. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2023.

Conforti, G., Durmus, A., Pham, L.-T.-N., and Raoul, G. Non-asymptotic convergence of discrete diffusion models: Masked and random walk dynamics. *arXiv preprint arXiv:2512.00580*, 2025.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dmitriev, D., Huang, Z., and Wei, Y. Efficient sampling with discrete diffusion models: Sharp and adaptive guarantees. *arXiv preprint arXiv:2602.15008*, 2026.

Gillespie, D. T. Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of chemical physics*, 115(4):1716–1733, 2001.

Gokaslan, A. and Cohen, V. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

Goldreich, O. Computational complexity: a conceptual perspective. *ACM Sigact News*, 2008.

Gruver, N., Stanton, S., Frey, N., Rudner, T. G., Hotzel, I., Lafrance-Vanasse, J., Rajpal, A., Cho, K., and Wilson, A. G. Protein design with guided discrete diffusion. *Advances in neural information processing systems*, 36:12489–12517, 2023.

Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., and Guo, B. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10696–10706, 2022.

Gupta, S., Cai, L., and Chen, S. Faster diffusion-based sampling with randomized midpoints: Sequential and parallel. *arXiv e-prints*, pp. arXiv–2406, 2024.

Huang, X., Lin, Y., Jain, N., Wang, K., Zou, D., Ma, Y., and Zhang, T. On the complexity theory of masked discrete diffusion: From $\mathrm{poly}(1/\epsilon)$ to nearly ϵ-free. *arXiv preprint arXiv:2509.21835*, 2025.

Kelly, F. P. *Reversibility and stochastic networks*. Cambridge University Press, 2011.

Lezama, J., Salimans, T., Jiang, L., Chang, H., Ho, J., and Essa, I. Discrete predictor-corrector diffusion models for image synthesis. In *The Eleventh International Conference on Learning Representations*, 2022.

Liang, Y., Huang, R., Lai, L., Shroff, N., and Liang, Y. Absorb and converge: Provable convergence guarantee for absorbing discrete diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Liang, Y., Liang, Y., Lai, L., and Shroff, N. Discrete diffusion models: Novel analysis and new sampler guarantees. *arXiv preprint arXiv:2509.16756*, 2025b.

Liang, Y., Tan, Z., Shroff, N., and Liang, Y. Sharp convergence rates for masked diffusion models. *arXiv preprint arXiv:2602.22505*, 2026. URL https://arxiv.org/abs/2602.22505.

Liu, S., Nam, J., Campbell, A., Stärk, H., Xu, Y., Jaakkola, T., and Gómez-Bombarelli, R. Think while you generate: Discrete diffusion with planned denoising. *arXiv preprint arXiv:2410.06264*, 2024.

Ou, J., Nie, S., Xue, K., Zhu, F., Sun, J., Li, Z., and Li, C. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*, 2024.

Peng, F. Z., Bezemek, Z., Patel, S., Rector-Brooks, J., Yao, S., Bose, A. J., Tong, A., and Chatterjee, P. Path planning for masked diffusion model sampling. *arXiv preprint arXiv:2502.03540*, 2025.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Razavi, A., Van den Oord, A., and Vinyals, O. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

Ren, Y., Chen, H., Rotskoff, G. M., and Ying, L. How discrete and continuous diffusion meet: Comprehensive analysis of discrete diffusion models via a stochastic integral framework. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Ren, Y., Chen, H., Zhu, Y., Guo, W., Chen, Y., Rotskoff, G. M., Tao, M., and Ying, L. Fast solvers for discrete diffusion models: Theory and applications of high-order algorithms. *arXiv preprint arXiv:2502.00234*, 2025b.

Ringel, L., Ali, A., and Romano, Y. Dependency-guided parallel decoding in discrete diffusion language models. *arXiv preprint arXiv:2604.02560*, 2026.

Sahoo, S. S., Deschenaux, J., Gokaslan, A., Wang, G., Chiu, J., and Kuleshov, V. The diffusion duality. *arXiv preprint arXiv:2506.10892*, 2025.

Sarkar, A., Kang, Y., Somia, N., Mantilla, P., Zhou, J. L., Nagai, M., Tang, Z., Zhao, C., and Koo, P. Designing dna with tunable regulatory activity using score-entropy discrete diffusion. *bioRxiv*, pp. 2024–05, 2024.

Shaul, N., Gat, I., Havasi, M., Severo, D., Sriram, A., Holderrieth, P., Karrer, B., Lipman, Y., and Chen, R. T. Flow matching with general discrete paths: A kinetic-optimal perspective. *arXiv preprint arXiv:2412.03487*, 2024.

Shih, A., Belkhale, S., Ermon, S., Sadigh, D., and Anari, N. Parallel sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:4263–4276, 2023.

Vignac, C., Krawczuk, I., Siraudin, A., Wang, B., Cevher, V., and Frossard, P. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022.

Wang, C., Uehara, M., He, Y., Wang, A., Biancalani, T., Lal, A., Jaakkola, T., Levine, S., Wang, H., and Regev, A. Fine-tuning discrete diffusion models via reward optimization with applications to dna and protein design. *arXiv preprint arXiv:2410.13643*, 2024.

Wu, C., Zhang, H., Xue, S., Liu, Z., Diao, S., Zhu, L., Luo, P., Han, S., and Xie, E. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *arXiv preprint arXiv:2505.22618*, 2025.

Xie, T., Xue, S., Feng, Z., Hu, T., Sun, J., Li, Z., and Zhang, C. Variational autoencoding discrete diffusion with enhanced dimensional correlations modeling. *arXiv preprint arXiv:2505.17384*, 2025.

Yang, D., Yu, J., Wang, H., Wang, W., Weng, C., Zou, Y., and Yu, D. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1720–1733, 2023.

Yu, L. and Dalalyan, A. Parallelized midpoint randomization for langevin monte carlo. *Stochastic Processes and their Applications*, pp. 104764, 2025.

Yu, R., Ma, X., and Wang, X. Dimple: Discrete diffusion multimodal large language model with parallel decoding. *arXiv preprint arXiv:2505.16990*, 2025.

Zhang, L. The cosine schedule is fisher-rao-optimal for masked discrete diffusion models. *arXiv preprint arXiv:2508.04884*, 2025.

Zhao, L., Ding, X., Yu, L., and Akoglu, L. Unified discrete diffusion for categorical data. *arXiv preprint arXiv:2402.03701*, 2024.

Zheng, K., Chen, Y., Mao, H., Liu, M.-Y., Zhu, J., and Zhang, Q. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. *arXiv preprint arXiv:2409.02908*, 2024.

Zheng, L., Yuan, J., Yu, L., and Kong, L. A reparameterized discrete diffusion model for text generation. *arXiv preprint arXiv:2302.05737*, 2023.

Zhou, H. and Sugiyama, M. Parallel simulation for sampling under isoperimetry and score-based diffusion models. *arXiv preprint arXiv:2412.07435*, 2024.

Zhou, H. and Sugiyama, M. Parallel simulation for log-concave sampling and score-based diffusion models. In *Forty-second International Conference on Machine Learning*, 2025.

Zhu, Y., Wang, X., Lathuiliere, S., and Kalogeiton, V. Dimo: Distilling masked diffusion models into one-step generator. *arXiv preprint arXiv:2503.15457*, 2025.

A. Related Work

A.1. Discrete Diffusion models

Masked and Uniformed discrete diffusion models have developed from a wide range of aspects. (Austin et al., 2021) provides foundational formalisms for discrete diffusion—multi-nomial corruption and structured transition matrices as the absorbing states. (Chao et al., 2025) introduce intermediate token states between masked/unmasked to avoid redundant computation when sequences barely change across steps. (Vignac et al., 2022) performs discrete denoising on graphs by noising/denoising categorical node and edge types. (Gu et al., 2022) code sequences with discrete diffusion for text-to-image, improving quality and speed versus auto-regressive token decoders. (Gruver et al., 2023) introduces NOS guidance to design protein sequences directly in sequence space, demonstrating antibody optimization in vitro. (Wang et al., 2024) optimizes discrete diffusion generators with task rewards to design biomolecular sequences.

A.2. Acceleration for Discrete Diffusion Sampling

So far, there have been numerous studies on accelerating sampling for discrete diffusion models. (Chen et al., 2024b) replaces the standard Markov chain with a non-Markov schedule to skip steps and cut the number of network calls without retraining. (Sahoo et al., 2025) adapts consistency distillation to discrete diffusion by constructing the duality connection between continuous and discrete diffusion. (Zhu et al., 2025) proposes a token initialization strategy that injects randomness while maintaining similarity to teacher training distribution, achieving one-step distillation of masked diffusion models. (Zheng et al., 2023; Ou et al., 2024) design equivalent reparameterization of discrete diffusion that yields more effective training and decoding strategies. (Wu et al., 2025) develops confidence-aware parallel decoding to accelerate multi-token sampling while maintaining accuracy. (Zhang, 2025) gives a principled choice of discretization schedule for efficient sampling. (Shaul et al., 2024) allows arbitrary discrete probability paths, giving more control to find shorter or easier trajectories with fewer steps for discrete generation. (Zhao et al., 2024) derives a simple backward denoising formula, enabling exact and accelerated sampling and unifying discrete-time and continuous-time discrete diffusion.

A.3. Picard Iteration

Parallel sampling based on Picard recursions has been applied to various models such as first-order Markov chains for both Monte Carlo methods (Yu & Dalalyan, 2025; Anari et al., 2024) and continuous generative models (Zhou & Sugiyama, 2024; Shih et al., 2023; Gupta et al., 2024).

Remark Although continuous diffusion and its parallel-in-time acceleration methods have already been well-established with detailed theoretical and empirical analysis, due to the significant gap of mathematical framework between discrete and continuous diffusion such as random source, dynamics and regularity condition, it is difficult to directly transfer continuous analysis tools and methods to the discrete case. Table 7 shows the main difference between discrete and continuous diffusion settings.

Table 7. Continuous vs. discrete diffusion from the viewpoint of Picard analysis.

Aspect	Continuous diffusion	Discrete diffusion
Randomness	Brownian motion	Poisson / jump events
State space	$\mathbb{R}^d$	$[S]^d$ or masking state space
Dynamics	SDE / ODE	CTMC / jump process
Reverse model	Reverse-time SDE / ODE	Reverse jump rates
Score	$\nabla_x \log p_t(x)$	Conditional token posterior
Regularity condition	Lipschitz score	Dobrushin event-switching bound

B. Proofs of results in Section 4.2

B.1. Proof of Theorem 4.6

We first introduce and prove some necessary lemmas.

Lemma B.1 (Global mass bound for loglinear masking)**.** *Assume $L_\star = O(1)$ and the loglinear masking rate satisfies*

$a(t) \asymp 1/t$ near the clean-data endpoint. Then

$$G_d \le L_\star \sum_{q \in \mathcal{Q}_d} \Delta_q a_q.$$

Moreover, for a sufficiently fine grid,

$$\sum_{q \in \mathcal{Q}_d} \Delta_q a_q = O\left(\int_\eta^T a(t)\mathrm{d}t\right) = O\left(\log \frac{T}{\eta}\right).$$

If T is at most logarithmic in d and $\eta \asymp \varepsilon/d$, then, up to lower-order logarithmic factors,

$$G_d = O\left(\log \frac{d}{\varepsilon}\right).$$

Proof. By Definition 4.1 and 4.3, we have

$$G_d = L_\star \sum_{q \in \mathcal{Q}_d} \rho_q \le L_\star \sum_{q \in \mathcal{Q}_d} \Delta_q a_q.$$

When the fine grid resolves the interval $[\eta, T]$, the last sum is a Riemann sum for the time-rate integral. Since $a(t) \asymp 1/t$ near the endpoint,

$$\sum_{q \in \mathcal{Q}_d} \Delta_q a_q = O\left(\int_\eta^T \frac{1}{t}\mathrm{d}t\right) = O\left(\log \frac{T}{\eta}\right).$$

Substituting $\eta \asymp \varepsilon/d$ gives $\log(T/\eta) = \log(Td/\varepsilon)$. If T grows at most logarithmically in d, the additional $\log T$ term is lower order compared with $\log(d/\varepsilon)$, which proves the claim. □

Fix a block and suppress its block index. Let the block contain M fine cells indexed by $r = 0, \dots, M-1$. Let $Z_m^{(k)}$ be the state at the m-th microstep after the k-th Picard trajectory has been constructed, where $m = 0, \dots, M$. The block start is fixed, so $Z_0^{(k)}$ is the block input for all k. Let

$$C_r^{(k)} = C_r(Z_r^{(k)}, \omega_r)$$

be the local proposal events generated in cell r from the k-th Picard input trajectory. Define the adjacent Picard residual

$$e_k(m) = \mathbb{E}\left[d_H\left(Z_m^{(k+1)}, Z_m^{(k)}\right)\right].$$

The endpoint is defined as $Y^{(k)} = Z_M^{(k)}$.

Lemma B.2 (First-hitting stability)**.** *Fix a block start state x. For a prefix of local proposal sets $C_{<m} = (C_0, \dots, C_{m-1})$, let $\Phi_m(C_{<m})$ be the state obtained from x by applying, for each coordinate, only its earliest proposed token in the prefix. Then, for any two prefixes $C_{<m}$ and $C'_{<m}$,*

$$d_H\big(\Phi_m(C_{<m}), \Phi_m(C'_{<m})\big) \le \sum_{r=0}^{m-1} |C_r \triangle C'_r|.$$

Proof. For coordinate i, write

$$C_r(i) = \{c \in [S] : (i, c) \in C_r\}.$$

By Definition 4.1, $C_r(i)$ contains at most one token. Let $\phi_i(C_{<m})$ be the i-th coordinate of $\Phi_m(C_{<m})$. Thus $\phi_i(C_{<m}) = x_i$ if $C_r(i) = \emptyset$ for every $r < m$, and otherwise $\phi_i(C_{<m})$ is the token in the earliest nonempty $C_r(i)$.

Define

$$\delta_i = \mathbf{1}\{\phi_i(C_{<m}) \ne \phi_i(C'_{<m})\}.$$

We first prove the coordinate-wise inequality

$$\delta_i \le \sum_{r=0}^{m-1} |C_r(i) \triangle C'_r(i)|.$$

If the right-hand side is zero, then every summand is zero, so $C_r(i) = C'_r(i)$ for all $r < m$. Hence the two coordinate-level proposal histories are identical. They either both contain no proposal for coordinate i, in which case both outputs equal x_i, or they have the same earliest nonempty cell and the same token there. In both cases $\phi_i(C_{<m}) = \phi_i(C'_{<m})$, so $\delta_i = 0$. This proves the coordinate-wise inequality.

Summing over coordinates gives

$$d_H\big(\Phi_m(C_{<m}), \Phi_m(C'_{<m})\big) = \sum_{i=1}^{d} \delta_i \le \sum_{i=1}^{d} \sum_{r=0}^{m-1} |C_r(i) \triangle C'_r(i)|.$$

Reordering the finite sums yields

$$\sum_{i=1}^{d} \sum_{r=0}^{m-1} |C_r(i) \triangle C'_r(i)| = \sum_{r=0}^{m-1} \sum_{i=1}^{d} |C_r(i) \triangle C'_r(i)|.$$

Since C_r is the disjoint union of the coordinate-level sets $\{i\} \times C_r(i)$,

$$\sum_{i=1}^{d} |C_r(i) \triangle C'_r(i)| = |C_r \triangle C'_r|.$$

Substituting this identity proves the lemma. □

Lemma B.3 (Triangular Picard recursion)**.** *Under Assumption 4.2, for every $m = 0, \dots, M$ and $k \ge 0$,*

$$e_{k+1}(m) \le \sum_{r=0}^{m-1} b_r e_k(r).$$

Proof. For $m = 0$, both sides are zero because the block start is the same for all Picard iterates. Let $m \ge 1$.

By construction, the $(k+2)$-nd Picard state at microstep m is obtained by applying first-hitting selection to the local proposals generated from the $(k+1)$-st Picard input trajectory. Therefore,

$$Z_m^{(k+2)} = \Phi_m(C_0^{(k+1)}, \dots, C_{m-1}^{(k+1)}).$$

Similarly,

$$Z_m^{(k+1)} = \Phi_m(C_0^{(k)}, \dots, C_{m-1}^{(k)}).$$

Applying Lemma B.2 to these two prefixes gives

$$d_H\left(Z_m^{(k+2)}, Z_m^{(k+1)}\right) \le \sum_{r=0}^{m-1} |C_r^{(k+1)} \triangle C_r^{(k)}|.$$

Taking expectations yields

$$e_{k+1}(m) \le \sum_{r=0}^{m-1} \mathbb{E}\left[|C_r^{(k+1)} \triangle C_r^{(k)}|\right].$$

For a fixed r, the two sets $C_r^{(k+1)}$ and $C_r^{(k)}$ are generated with the same cell randomness ω_r from inputs $Z_r^{(k+1)}$ and $Z_r^{(k)}$. Conditioning on these two inputs and using Assumption 4.2,

$$\mathbb{E}\left[|C_r^{(k+1)} \triangle C_r^{(k)}| \mid Z_r^{(k+1)}, Z_r^{(k)}\right] \le L_\star \rho_r d_H\left(Z_r^{(k+1)}, Z_r^{(k)}\right).$$

Taking expectation again and using $b_r = L_\star \rho_r$ gives

$$\mathbb{E}\left[|C_r^{(k+1)} \triangle C_r^{(k)}|\right] \le b_r e_k(r).$$

Substituting this bound into the previous display proves the recursion. □

Now we restate Theorem 4.6 and start the main proof.

Theorem B.4 (Picard endpoint convergence)**.** *Let*

$$B = \sum_{r=0}^{M-1} b_r, \qquad E_0 = \max_{0 \le m \le M} e_0(m).$$

Under Assumption 4.2, the block endpoint iterates converge in expected Hamming distance to a limit $Y^{(\infty)}$, and for every $K \ge 0$,

$$\mathbb{E}\left[d_H\left(Y^{(K)}, Y^{(\infty)}\right)\right] \le E_0 e^B \frac{B^K}{K!}.$$

For block n, the same statement holds with $B = B_n$. One may always take $E_0 \le d$.

Proof. We first prove a bound on adjacent endpoint residuals. We claim that for every $K \ge 0$,

$$e_K(M) \le E_0 \frac{B^K}{K!}.$$

For $K = 0$, this is immediate from the definition of E_0. Now let $K \ge 1$. Repeatedly applying Lemma B.3 gives

$$e_K(M) \le \sum_{r_1 < M} b_{r_1} e_{K-1}(r_1).$$

Applying the recursion to $e_{K-1}(r_1)$ gives

$$e_{K-1}(r_1) \le \sum_{r_2 < r_1} b_{r_2} e_{K-2}(r_2).$$

Substituting,

$$e_K(M) \le \sum_{r_2 < r_1 < M} b_{r_1} b_{r_2} e_{K-2}(r_2).$$

Continuing this expansion for K steps yields

$$e_K(M) \le \sum_{r_K < r_{K-1} < \cdots < r_1 < M} b_{r_1} \cdots b_{r_K} e_0(r_K).$$

Since $e_0(r_K) \le E_0$,

$$e_K(M) \le E_0 S_K, \qquad S_K := \sum_{r_K < \cdots < r_1 < M} b_{r_1} \cdots b_{r_K}.$$

It remains to bound S_K. Expanding B^K gives

$$B^K = \left(\sum_{r=0}^{M-1} b_r\right)^K = \sum_{(i_1,\ldots,i_K) \in \{0,\ldots,M-1\}^K} b_{i_1} \cdots b_{i_K}.$$

All terms are nonnegative. Hence B^K is at least the sub-sum over tuples with K distinct indices. For each unordered subset $\{a_1, \ldots, a_K\}$ of K distinct indices, the product $\prod_a b_a$ appears exactly $K!$ times in the distinct-index tuple expansion, once for each permutation. The strictly decreasing chains in S_K are in one-to-one correspondence with these unordered subsets. Therefore

$$B^K \ge K! S_K, \qquad S_K \le \frac{B^K}{K!}.$$

Thus

$$e_K(M) \le E_0 \frac{B^K}{K!}.$$

Now fix $L > K$. By the triangle inequality for Hamming distance,

$$d_H(Y^{(L)}, Y^{(K)}) \le \sum_{\ell=K}^{L-1} d_H(Y^{(\ell+1)}, Y^{(\ell)}).$$

Taking expectations and using $Y^{(\ell)} = Z_M^{(\ell)}$ gives

$$\mathbb{E}[d_H(Y^{(L)}, Y^{(K)})] \le \sum_{\ell=K}^{L-1} e_\ell(M) \le E_0 \sum_{\ell=K}^{L-1} \frac{B^\ell}{\ell!}.$$

The exponential series converges, so the right-hand side tends to zero as $K, L \to \infty$. Hence the endpoints are Cauchy in expected Hamming distance and have a limit, denoted $Y^{(\infty)}$.

Letting $L \to \infty$ in the previous inequality gives

$$\mathbb{E}[d_H(Y^{(\infty)}, Y^{(K)})] \le E_0 \sum_{\ell=K}^{\infty} \frac{B^\ell}{\ell!}.$$

Finally,

$$\sum_{\ell=K}^{\infty} \frac{B^\ell}{\ell!} = \frac{B^K}{K!} \sum_{j=0}^{\infty} \frac{B^j K!}{(K+j)!}.$$

Since $K!j! \le (K+j)!$, we have $K!/(K+j)! \le 1/j!$. Hence

$$\sum_{j=0}^{\infty} \frac{B^j K!}{(K+j)!} \le \sum_{j=0}^{\infty} \frac{B^j}{j!} = e^B.$$

Therefore

$$\mathbb{E}[d_H(Y^{(\infty)}, Y^{(K)})] \le E_0 e^B \frac{B^K}{K!}.$$

This completes the proof. □

B.2. Proof of Corollary 4.8

We restate Corollary 4.8 for reference and start the main proof.

Corollary B.5 (Uniform-block Picard NFE)**.** *Let $H := T - \eta$ be the early-stopped sampling horizon. Choose a constant physical block width $h_0 > 0$ independent of d and ε_{pic}, and partition $[\eta, T]$ into*

$$N = \left\lceil \frac{H}{h_0} \right\rceil$$

uniform physical-time blocks, with the last block possibly shorter. Under Assumption 4.2, given $T = O(\log(d \log S/\varepsilon_{\text{pic}}))$, let $B_{\max} = \max_n B_n$. Then $B_{\max} \le G_d$, and the choice

$$K_p = \left\lceil \max\left\{ 2eB_{\max}, \frac{1}{\log 2} \log \frac{Nde^{B_{\max}}}{\varepsilon_{\text{pic}}} \right\} \right\rceil$$

ensures that the Picard endpoint TV error is at most ε_{pic}. Consequently,

$$K_p = O\left(\log \frac{d}{\varepsilon_{\text{pic}}} \right), \qquad \text{NFE}_{\text{Picard}} = NK_p = O\left(\log \frac{d}{\varepsilon_{\text{pic}}} \cdot \log \frac{d \log S}{\varepsilon_{\text{pic}}} \right).$$

Proof. The role of the block count is to keep the physical width of each Picard block uniformly bounded. Since the early-stopped horizon has length $H = T - \eta$, choosing

$$N = \left\lceil \frac{H}{h_0} \right\rceil$$

implies that every non-final block has width h_0 and the final block has width at most h_0. Thus all block widths are $O(1)$. Since h_0 is a constant and $\eta \le T$,

$$N \le \frac{T}{h_0} + 1 = O(T).$$

The assumption $T = O(\log(d \log S/\varepsilon_{\mathrm{pic}}))$ therefore gives

$$N = O\left(\log \frac{d \log S}{\varepsilon_{\mathrm{pic}}}\right).$$

For block n, Theorem 4.6 gives

$$\mathbb{E}[d_H(Y_n^{(K_p)}, Y_n^{(\infty)})] \le E_{n,0} e^{B_n} \frac{B_n^{K_p}}{K_p!}.$$

By coupling the Picard and fixed-point block endpoints using the same block randomness,

$$\mathrm{TV}(\mathcal{L}(Y_n^{(K_p)}), \mathcal{L}(Y_n^{(\infty)})) \le \mathbb{E}[d_H(Y_n^{(K_p)}, Y_n^{(\infty)})].$$

Using the first block where the two coupled chains disagree and then applying a union bound over blocks,

$$E_{\mathrm{pic}} \le \sum_{n=1}^{N} E_{n,0} e^{B_n} \frac{B_n^{K_p}}{K_p!}.$$

Since $E_{n,0} \le d$ and $B_n \le B_{\max}$,

$$E_{\mathrm{pic}} \le N d e^{B_{\max}} \frac{B_{\max}^{K_p}}{K_p!}.$$

It is enough to make the last expression at most $\varepsilon_{\mathrm{pic}}$.

Using $K! \ge (K/e)^K$,

$$\frac{B_{\max}^K}{K!} \le \left(\frac{eB_{\max}}{K}\right)^K.$$

Thus it suffices that

$$N d e^{B_{\max}} \left(\frac{eB_{\max}}{K}\right)^K \le \varepsilon_{\mathrm{pic}}.$$

Equivalently,

$$\left(\frac{K}{eB_{\max}}\right)^K \ge \frac{N d e^{B_{\max}}}{\varepsilon_{\mathrm{pic}}},$$

or

$$K \log\left(\frac{K}{eB_{\max}}\right) \ge \log \frac{N d e^{B_{\max}}}{\varepsilon_{\mathrm{pic}}}.$$

If $K \ge 2eB_{\max}$, then $\log(K/(eB_{\max})) \ge \log 2$. If also

$$K \ge \frac{1}{\log 2} \log \frac{N d e^{B_{\max}}}{\varepsilon_{\mathrm{pic}}},$$

then the previous inequality holds. This proves the displayed sufficient choice of K_p.

It remains to simplify the order. Since the blocks partition the fine grid,

$$B_{\max} \le \sum_{n=1}^{N} B_n = G_d.$$

By Lemma B.1,

$$B_{\max} = O\left(\log \frac{d}{\varepsilon_{\text{pic}}}\right), \qquad N = O\left(\log \frac{d \log S}{\varepsilon_{\text{pic}}}\right).$$

The first term in the maximum defining K_p is therefore $O(\log(d/\varepsilon_{\text{pic}}))$. For the second term,

$$\log \frac{N d e^{B_{\max}}}{\varepsilon_{\text{pic}}} = \log N + \log \frac{d}{\varepsilon_{\text{pic}}} + B_{\max}.$$

The term $\log N$ is lower order because N is logarithmic, and $B_{\max}$ has the same logarithmic order. Hence the second term is also $O(\log(d/\varepsilon_{\text{pic}}))$. Therefore

$$K_p = O\left(\log \frac{d}{\varepsilon_{\text{pic}}}\right).$$

Combining this with $N = O(\log(d/\varepsilon_{\text{pic}}))$ gives

$$\text{NFE}_{\text{Picard}} = NK_p = O\left(\log \frac{d}{\varepsilon_{\text{pic}}} \cdot \log \frac{d \log S}{\varepsilon_{\text{pic}}}\right).$$

□

B.3. Proof of the exponential-factorial contraction in continuous diffusion

Consider the integral equation usually used in continuous diffusion analysis:

$$X(t) = X_0 + \int_0^t F(s, X(s))\, ds, \qquad 0 \le t \le T,$$

and its Picard iteration

$$X^{(k+1)}(t) = X_0 + \int_0^t F(s, X^{(k)}(s))\, ds.$$

Assume that F is Lipschitz in its state argument with a time-dependent coefficient $L(s)$, namely

$$\|F(s, x) - F(s, y)\| \le L(s)\|x - y\|.$$

Define the adjacent Picard error

$$e_k(t) := \|X^{(k+1)}(t) - X^{(k)}(t)\|.$$

Then

$$e_{k+1}(t) \le \int_0^t L(s) e_k(s)\, ds.$$

Indeed, subtracting two consecutive Picard updates gives

$$X^{(k+2)}(t) - X^{(k+1)}(t) = \int_0^t \left[F(s, X^{(k+1)}(s)) - F(s, X^{(k)}(s))\right] ds.$$

Taking norms and applying the Lipschitz condition yields

$$e_{k+1}(t) = \|X^{(k+2)}(t) - X^{(k+1)}(t)\| \le \int_0^t L(s)\|X^{(k+1)}(s) - X^{(k)}(s)\|\, ds = \int_0^t L(s) e_k(s)\, ds.$$

Iterating this Volterra inequality gives an ordered-time expansion. Let

$$B_T := \int_0^T L(s)\, ds$$

and assume $e_0(t) \le E_0$ for all $t \in [0, T]$. For $K = 1$,

$$e_1(T) \le \int_0^T L(s_1) e_0(s_1)\, ds_1.$$

For $K = 2$,

$$e_2(T) \le \int_0^T L(s_1) \int_0^{s_1} L(s_2) e_0(s_2)\, ds_2\, ds_1.$$

Repeating this argument yields

$$e_K(T) \le E_0 \int_{0<s_K<\cdots<s_1<T} \prod_{\ell=1}^{K} L(s_\ell)\, ds_K \cdots ds_1.$$

The ordered simplex $\{0 < s_K < \cdots < s_1 < T\}$ is one of the $K!$ equal-ordering regions of $[0,T]^K$. Since the integrand $\prod_{\ell=1}^{K} L(s_\ell)$ is symmetric in the variables $s_1, \ldots, s_K$, we have

$$\int_{0<s_K<\cdots<s_1<T} \prod_{\ell=1}^{K} L(s_\ell)\, ds_K \cdots ds_1 = \frac{1}{K!} \left(\int_0^T L(s)\, ds \right)^K = \frac{B_T^K}{K!}.$$

Therefore,

$$e_K(T) \le E_0 \frac{B_T^K}{K!}.$$

If $X^\star$ denotes the fixed point of the integral equation, then the fixed-point error can be controlled by summing the adjacent errors:

$$\|X^\star(T) - X^{(K)}(T)\| \le \sum_{\ell=K}^{\infty} e_\ell(T) \le E_0 \sum_{\ell=K}^{\infty} \frac{B_T^\ell}{\ell!}.$$

Using the elementary tail bound

$$\sum_{\ell=K}^{\infty} \frac{B_T^\ell}{\ell!} \le e^{B_T} \frac{B_T^K}{K!},$$

we obtain

$$\|X^\star(T) - X^{(K)}(T)\| \le E_0 e^{B_T} \frac{B_T^K}{K!}.$$

For reverse SDEs with state-independent diffusion coefficient, the same argument applies pathwise when all Picard iterates are driven by the same Brownian path with fixed random seeds, since the stochastic integral term cancels after subtracting two consecutive iterates. Hence, the factor $B^K/K!$ in our discrete first-hit Picard analysis should be viewed as the time-discrete version of the classical ordered-simplex factor in continuous Picard theory.

C. Proofs of results in Section 4.3

C.1. Proof of Proposition 4.9

We restate Proposition 4.9 for reference and start the main proof.

Proposition C.1 (Total TV error with absorbing τ-leaping reference)**.** *Assume the absorbing serial τ-leaping bound above and the Picard endpoint bound in Theorem 4.6. Then*

$$\mathrm{TV}(\mu_0, A_\eta \# \nu_{\mathrm{pic}}^{\theta,\Delta,K_p}) \le E_{\mathrm{term}} + \sqrt{\frac{1}{2} E_{\mathrm{absTL}}} + \sum_{n=1}^{N} E_{n,0} e^{B_n} \frac{B_n^{K_p}}{K_p!}.$$

Proof. Insert the two intermediate distributions $A_\eta \# \mu_\eta^\star$ and $A_\eta \# \nu_{\mathrm{seq}}^{\theta,\Delta}$. By the triangle inequality for total variation,

$$\begin{aligned} \mathrm{TV}(\mu_0, A_\eta \# \nu_{\mathrm{pic}}^{\theta,\Delta,K_p}) \le\ & \mathrm{TV}(\mu_0, A_\eta \# \mu_\eta^\star) \\ & + \mathrm{TV}(A_\eta \# \mu_\eta^\star, A_\eta \# \nu_{\mathrm{seq}}^{\theta,\Delta}) \\ & + \mathrm{TV}(A_\eta \# \nu_{\mathrm{seq}}^{\theta,\Delta}, A_\eta \# \nu_{\mathrm{pic}}^{\theta,\Delta,K_p}). \end{aligned}$$

The first term is E_{term}. Since deterministic maps cannot increase total variation,

$$\mathrm{TV}(A_\eta \# P, A_\eta \# Q) \le \mathrm{TV}(P, Q),$$

the second term is bounded by $\mathrm{TV}(\mu_\eta^\star, \nu_{\mathrm{seq}}^{\theta,\Delta})$. Pinsker's inequality and the absorbing serial KL bound give

$$\mathrm{TV}(\mu_\eta^\star, \nu_{\mathrm{seq}}^{\theta,\Delta}) \le \sqrt{\frac{1}{2}\mathrm{KL}(\mu_\eta^\star \| \nu_{\mathrm{seq}}^{\theta,\Delta})} \le \sqrt{\frac{1}{2}E_{\mathrm{absTL}}}.$$

For the third term, couple the serial and Picard samplers by using the same block random sources. For any coupled random variables X and Y,

$$\mathrm{TV}(\mathcal{L}(X), \mathcal{L}(Y)) \le \mathbb{P}(X \ne Y) \le \mathbb{E}[d_H(X, Y)].$$

Applying this relation to the first block where the coupled endpoints differ and then using a union bound over blocks gives

$$\mathrm{TV}(\nu_{\mathrm{seq}}^{\theta,\Delta}, \nu_{\mathrm{pic}}^{\theta,\Delta,K_p}) \le \sum_{n=1}^{N} E_{n,0} e^{B_n} \frac{B_n^{K_p}}{K_p!},$$

by Theorem 4.6. Combining the displayed inequalities proves the claim. □

C.2. Proof of Theorem 4.10

We restate Theorem 4.10 for reference and start the main proof.

Theorem C.2 (Global complexity)**.** *Let the target total variation error be $\varepsilon_{\mathrm{tot}}$. Under Assumption 4.2, 4.5, 4.4, suppose the score error is controlled at order $\varepsilon_{\mathrm{tot}}^2$, then choosing*

$$\eta = \Theta(\varepsilon_{\mathrm{tot}}/d), \qquad T = O\left(\log \frac{d \log S}{\varepsilon_{\mathrm{tot}}^2}\right), \qquad N_{\mathrm{fine}} = \widetilde{O}\left(\frac{dS}{\varepsilon_{\mathrm{tot}}^2}\right)$$

controls the serial and early-stopping terms. With constant physical block width, the number of blocks and Picard depth with

$$N_{\mathrm{block}} = O\left(\log \frac{d \log S}{\varepsilon_{\mathrm{tot}}^2}\right), \qquad K_p = O\left(\log \frac{d}{\varepsilon_{\mathrm{tot}}}\right)$$

makes the Picard term of order $\varepsilon_{\mathrm{tot}}$. Consequently, the global time and space complexity are

$$N_{\mathrm{block}} K_p = O\left(\log \frac{d \log S}{\varepsilon_{\mathrm{tot}}^2} \cdot \log \frac{d}{\varepsilon_{\mathrm{tot}}}\right), \qquad O\left(d \frac{N_{\mathrm{fine}}}{N}\right) = \widetilde{O}\left(\frac{d^2 S}{\varepsilon_{\mathrm{tot}}^2}\right)$$

Proof. The early-stopping term is $O(d\eta)$, so $\eta = \Theta(\varepsilon_{\mathrm{tot}}/d)$ makes it $O(\varepsilon_{\mathrm{tot}})$. The serial contribution enters the total TV bound through Pinsker, so we require

$$E_{\mathrm{absTL}} = O(\varepsilon_{\mathrm{tot}}^2).$$

The initialization part $de^{-T} \log S$ is then controlled by taking

$$T = O\left(\log \frac{d \log S}{\varepsilon_{\mathrm{tot}}^2}\right).$$

The discretization part of E_{absTL} is

$$\frac{dS\big(T + \log(M_{\mathrm{score}}\eta^{-1})\big)\big(T + \log \eta^{-1}\big)^2}{N_{\mathrm{fine}}}.$$

Thus it is at most $O(\varepsilon_{\mathrm{tot}}^2)$ whenever

$$N_{\mathrm{fine}} = \widetilde{O}\left(\frac{dS}{\varepsilon_{\mathrm{tot}}^2}\right),$$

where logarithmic factors in d, $1/\eta$, $1/\varepsilon_{\mathrm{tot}}$, and M_{score} are hidden.

For the Picard part, choosing a constant physical block width gives $N_{\text{block}} = O(T)$. Moreover, Lemma B.1 yields

$$B_{\max} \le G_d = O\left(\log \frac{d}{\varepsilon_{\text{tot}}}\right).$$

The safe depth choice in Corollary 4.8 then gives

$$K_p = O\left(\log \frac{d}{\varepsilon_{\text{tot}}}\right).$$

Multiplying this by $N_{\text{block}} = O(T)$ proves the claimed NFE bound. □

D. Algorithm of the first-hitting truncation

Algorithm 2 provides details about how our first-hitting truncation is implemented during the Picard sampling. We treat this first-hitting mechanism as a single abstract operation in the theory. Given the local proposal events in a block prefix, the first-hitting operation returns the state obtained by applying, for each coordinate, only its earliest proposed token. This abstraction keeps the analysis focused on the only property needed for convergence: first-hitting does not amplify event-level discrepancies.

Algorithm 2 First-Hitting Truncation (FHT)

Input: Block-start state $\widehat{y}_{t_n}$ and jump sequence $\Delta\widehat{y}_0^{(k)}, \dots, \Delta\widehat{y}_{M-1}^{(k)}$

Output: Truncated jump sequence $\Delta\widehat{y}_0^{(k)}, \dots, \Delta\widehat{y}_{M-1}^{(k)}$

1: For all coordinates i and microsteps j, define $h_{j,i} \leftarrow \mathbf{1}\{\Delta\widehat{y}_{j,i}^{(k)} \neq 0\}$.

2: Compute the prefix counts $s_{j,i} \leftarrow \sum_{\ell=0}^{j} h_{\ell,i}$ for all i, j using a parallel prefix scan over j.

3: For all i, j in parallel, set $\Delta\widehat{y}_{j,i}^{(k)} \leftarrow \Delta\widehat{y}_{j,i}^{(k)}\, \mathbf{1}\{\widehat{y}_{t_n,i} = \text{MASK}\}\, \mathbf{1}\{s_{j,i} = 1\}$. **return** $\Delta\widehat{y}_0^{(k)}, \dots, \Delta\widehat{y}_{M-1}^{(k)}$

E. Extra Experiment Results and Details

In this section, we provide more details and results of the experiments.

E.1. Assumption Verification

In this section, we empirically verify Assumption 4.2. To assess the normalized event-switching sensitivity $L_\star$ along actual generation trajectories, we compute an on-trajectory weighted constant estimator $L_d^{\text{switch}} = (\sum_q U_q)/(\sum_q \rho_q D_q)$. Here U_q is the cumulative candidate-event set difference at fine cell q, D_q is the cumulative Hamming difference between adjacent Picard inputs at that cell, and ρ_q is the one-proposal probability. Equivalently, L_d^{switch} is a weighted average of the local ratios $L_q = U_q/(\rho_q D_q)$ with weights proportional to $\rho_q D_q$. Since $\rho_q D_q$ is the exposure of cell q in the Picard error recursion, this estimator emphasizes the cells that contribute most to aggregate iteration error.

Empirically, as shown in Figure 2, L_d^{switch} remains between 2.31 and 2.50 for $d \in [128, 640]$, with fitted log-log slope -0.043, supporting the assumption that the effective normalized sensitivity is dimension-independent along generation trajectories.

E.2. Synthetic data

In this section we provide more details about synthetic experiments.

E.2.1. 2D Toy Model

Chessboard The chessboard distribution has several key characteristics that make it an excellent test benchmark: (1) Discrete: The distribution is defined on a finite set of points on a two-dimensional grid; (2) Sparse: Nearly half of the grid points have a probability of exactly zero. A successful sampler must learn to restrict its generated samples strictly to the points that have non-zero probability mass (i.e., the support of the distribution); (3) Multi-modal: The probability

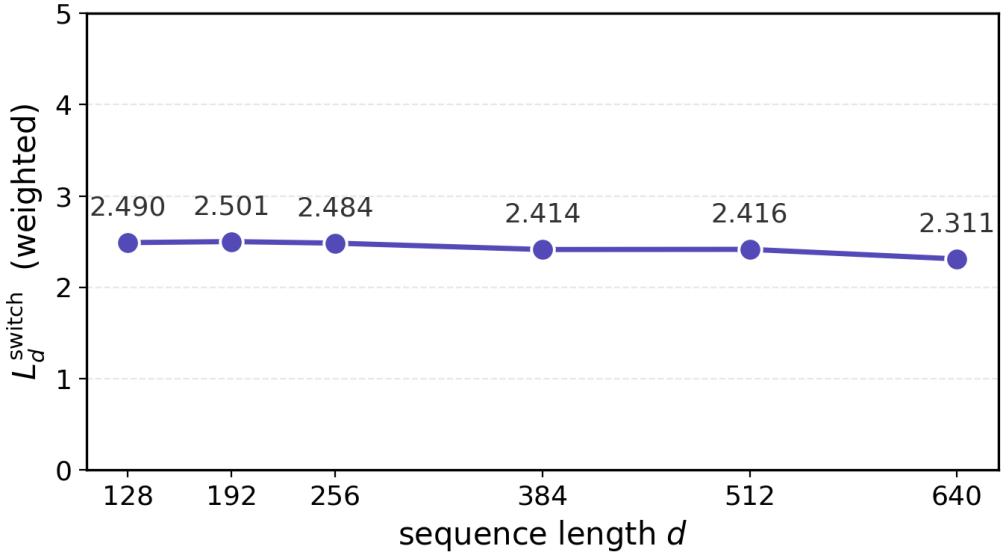


Figure 2. Visualization of $L_\star$.

mass is distributed across multiple, disconnected modes rather than being concentrated in a single region. The sampler is required to capture all of these distinct modes; (4) Structured: It exhibits a distinct, non-random geometric structure. This challenges the sampler's ability to reproduce the correct global pattern, rather than merely matching general statistical moments.

We conduct the experiment on a 8×8 chessboard distribution with varying Picard iteration depths K_p. We fix $N = 40, M = 80$ with totally 4096 samples. Runtimes and KL Divergence are averaged over 20 runs.

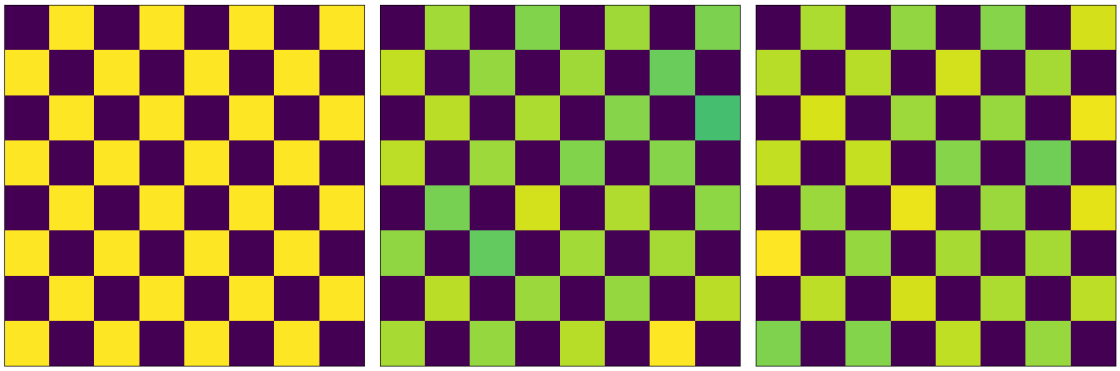


Figure 3. Visualization of the chessboard experiments. (Left) The target distribution. (Middle) The Picard sampling result. (Right) The sequential sampling result.

Circle The ring distribution on a 2D discrete grid concentrates its entire probability mass on grid points located within an annulus defined by an inner radius r_{in} and an outer radius r_{out}. A key characteristic is its Non-Convexity; the high-probability region encloses a central "hole" of zero probability, which makes the distribution's support non-convex. Another defining feature is its Connectivity. Unlike the disjoint, multi-modal structure of the checkerboard distribution, the support of the ring distribution forms a single connected component, meaning any point on the ring can traverse to any other point through a series of steps to adjacent locations.

We conduct the experiment on a circle distribution at 32×32 2D grid with varying Picard iteration depths K_p. We fix $N = 40, M = 80$ with totally 4096 samples. Runtime and KL Divergence are averaged over 20 runs.

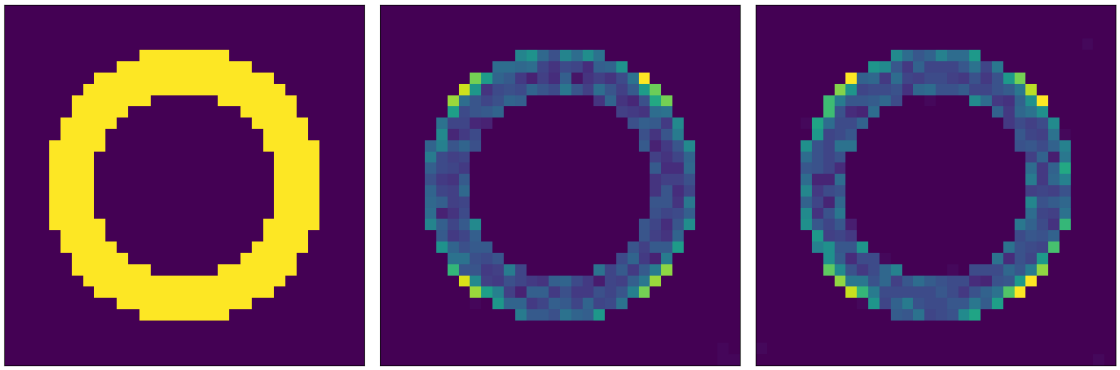


Figure 4. Visualization of the circle experiments. (Left) The target distribution. (Middle) The Picard sampling result. (Right) The sequential sampling result.

E.2.2. DIMENSIONAL SCALING

Oracle target and metrics. The oracle conditional probabilities are computed exactly from the partially unmasked group state to remove score approximation error. Since each sequence contains d/g independent groups, reliable group-level statistics can be obtained with a small batch size; we use only 8 samples for every d to avoid large-batch bandwidth effects. Quality is measured on the empirical group distribution. We report the per-coordinate group KL together with group TV and off-mode mass during schedule selection. The serial reference uses an oracle masked tau-leaping fine grid with $N_{\text{fine}}(d) \approx d$. For the Picard sampler we fix $K_p = 2$ and search over logarithmic block schedules, selecting the smallest block count whose sampling quality remain within prescribed margins of the serial reference. More visualization results are in Figure 5.

E.3. Real-world data

In this section we provide more results and samples for text/image generation tasks. The real-data experiment code is built based on the open-source codebase of (Ren et al., 2025b).

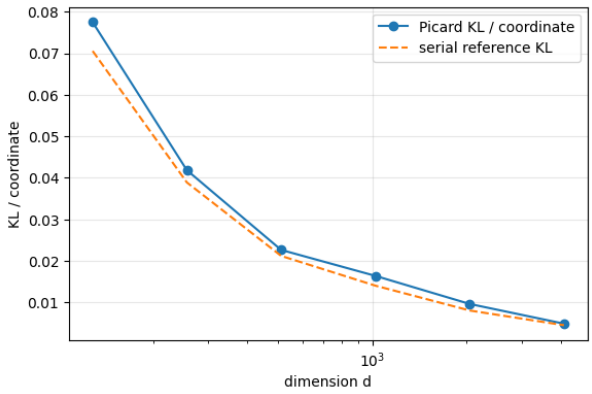
Figure 6 demonstrates sample images generated by our parallel method. Table 8 shows the comparison results between DDPD (Liu et al., 2024) and our parallel method.

Table 9 demonstrates the GPU memory cost under certain parameter settings for both image and text generation tasks.

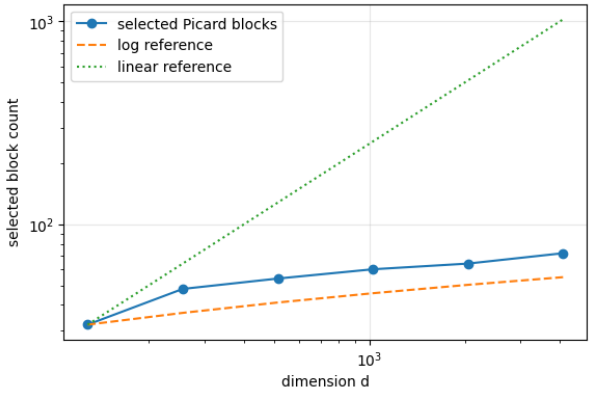
We would like to mention that, the MaskGIT we used in the image experiment utilizes the VQVAE (Razavi et al., 2019) to compress images into a discrete latent space with only 1024 codebook size and 16*16 sequence length, while the RADD for text generation works in a raw high-dimensional token space. Therefore, MaskGIT has much lower codebook and sequence dimensions, which can significantly decrease the active memory cost for each token sequence in the transformer. What's more, a large portion of memory for MaskGIT is likely occupied by static model weights, which do not scale with the parallel width M. The dynamic increase from parallelization is a small fraction of the total footprint.

Table 8. Generative perplexity of texts generated by DDPD and Picard τ-leaping

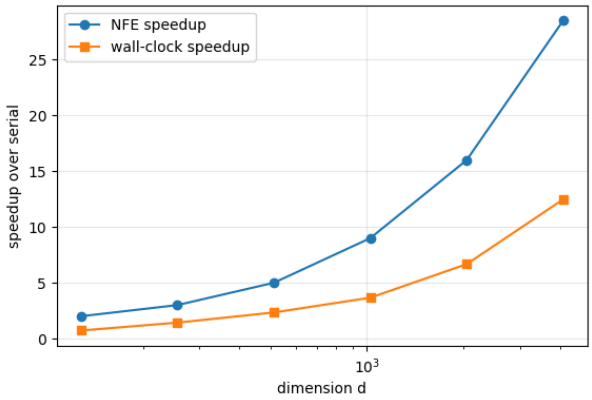
Method	**NFE**	**Perplexity**
DDPD-small softmax	1024	41.342
DDPD-medium softmax	1024	34.166
DDPD-small sigmoid	1024	30.587
DDPD-medium sigmoid	1024	28.025
Ours	512	25.876



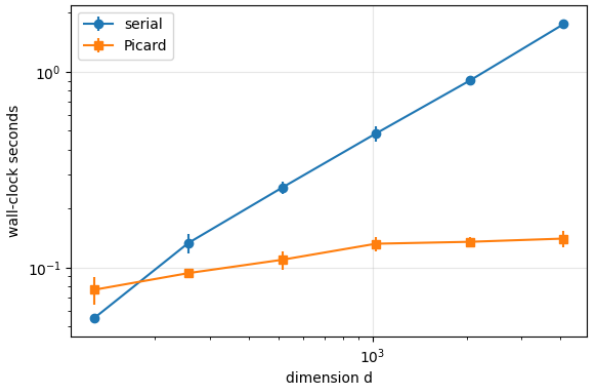
(a) Quality under selected schedule



(b) Selected Picard block count



(c) NFE and wall-clock speedup



(d) Serial vs. Picard wall-clock

Figure 5. Oracle synthetic quality-matched scaling experiment. Panel (a) compares the per-coordinate group KL of the selected Picard schedule with the serial oracle reference. Panel (b) shows that the selected number of Picard blocks grows much more slowly than the linear reference. Panel (c) compares the critical-path NFE speedup with the measured wall-clock speedup. Panel (d) reports the absolute wall-clock sampling time of the serial and Picard samplers.

Table 9. Parameter settings and GPU memory cost for image and text generation.

Tasks	**Seq. Cost**	**Para. Cost**	**M**	**N**	K_p
Text	6.7GB	11.6GB	8	32	2
Image	3.0GB	3.7GB	10	20	2

Figure 6. Generated samples from the Imagenet experiments.